



**SUBIECTUL al II-lea**

**(30 de puncte)**

**Pentru fiecare dintre itemii 1 și 2 scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Fiind dat un tablou bidimensional cu 20 linii și 20 coloane, câte elemente se găsesc strict deasupra diagonalei secundare a tabloului? **(4p.)**  
a. 190                                      b. 400                                      c. 180                                      d. 200
2. Se consideră o stivă în care au fost introduse, în această ordine, numerele 18, 62 și 44. Care dintre valorile din stivă va fi eliminată prima? **(4p.)**  
a. 18                                      b. 62                                      c. 44                                      d. oricare dintre ele

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

3. Se consideră declarațiile alăturate, în care variabila segment memorează coordonatele (abscisa și ordonata), în sistemul de coordonate  $xOy$ , ale extremităților unui segment. Scrieți o expresie C/C++ care are valoarea 1 dacă și numai dacă ambele extremități ale segmentului aparțin axei  $Oy$  a sistemului de coordonate.

```
struct punct
{
    int ordonata, abcisa;
};
```

```
struct
{
    punct A,B;
}segment;
```

**(6p.)**

4. Se consideră două șiruri de caractere  $a$ , de lungime  $na$  și  $b$ , de lungime  $nb$ . Se cere să se scrie o secvență de program C/C++ care determină dacă cele două șiruri de caractere conțin aceleași litere. În caz afirmativ, să se afișeze literele comune, o singură dată fiecare literă.

**Exemplu:** *marca* și *ammrc* conțin aceleași litere. Literele comune vor fi afișate în ordine alfabetică: *a c m r*. **(6p.)**

5. Scrieți un program C/C++ care citește de la tastatură un număr natural  $n$  ( $2 < n \leq 50$ ) și construiește în memorie un tablou bidimensional cu  $n$  linii și  $n$  coloane în care:

- elementele aflate pe pătrate concentrice au aceeași valoare, egală cu cea a numărului de elemente de pe pătratul concentric.

Programul afișează pe ecran tabloul obținut, fiecare linie a tabloului pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spațiu.

**Exemplu:** pentru  $n=6$  pe ecran se afișează tabloul alăturat.

```
20  20  20  20  20  20
20  12  12  12  12  20
20  12  4   4   12  20
20  12  4   4   12  20
20  12  12  12  12  20
20  20  20  20  20  20
```

**(10p.)**

**SUBIECTUL al III-lea**

**(30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Directorul școlii trebuie să programeze cele 3 probe de simulare a examenului de bacalaureat pe parcursul a 5 zile de școală. În câte moduri poate realiza această programare, știind că nu este permisă decât programarea unei singure probe într-o zi. **(4p.)**
- a. 15                                      b. 10                                      c. 27                                      d. 60

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră subprogramul  $f$ , definit alăturat. Ce se afișează pe ecran la apelul:  
`cout<<' '<<f(8174);?`

**(6p.)**

```
int f(int x)
{
    int n;
    if(x>0)
    {   if(x%2==0)
        {   cout<<x%10; | printf("%d",x%10);
            n=1+f(x/10);
        }else
        {   n=1+f(x/10);
            cout<<x%10; | printf("%d",x%10);
        }
        return n;
    }
    else return 0;
}
```

3. Fișierul text **numere.in** conține, pe prima linie un număr natural  $n$  ( $n \leq 100$ ), pe al doilea rând  $n$  numere naturale nenule de cel mult patru cifre fiecare, iar pe al treilea rând un număr natural  $a$  de cel mult 4 cifre. Scrieți programul C/C++ care creează fișierul text **numere.out** și scrie în el, pe prima linie, separate prin câte un spațiu numerele de pe a doua linie din fișier care sunt prime cu  $a$ . Dacă nu există niciun astfel de număr, atunci în fișierul text **numere.out** se va scrie mesajul **nu există**.

**Exemplu:**

**numere.in**

6  
27 25 15 7 45 9  
12

**numere.out**

25 7

**(10p.)**

4. Fie două mulțimi  $A$  și  $B$ , fiecare având cel mult 100 de elemente numere naturale de cel mult 4 cifre fiecare. Un element  $x \in A$  este numit **majorant** al mulțimii  $B$  dacă  $y \leq x$ , oricare ar fi  $y \in B$ . Fișierele text **A.txt** și **B.txt** conțin fiecare, separate prin câte un spațiu, cel mult 100 de elemente numere naturale distincte, fiecare având cel mult 4 cifre. Să se scrie un program care citește numerele din cel două fișiere și afișează pe ecran mesajul „**Există număr majorant**” sau mesajul „**Nu există număr majorant**”, după cum în fișierul text **A.txt** există, sau nu există cel puțin un număr majorant. Alegeți un algoritm eficient din punct de vedere al memoriei utilizate și al timpului de executare.

**Exemplu:** pentru următoarele fișiere

**A.txt**

8 3 11 26 4 19 10

**B.txt**

7 19 25 4 2

se va afișa **Există număr majorant**

- a. Descrieți în limbaj natural o metodă eficientă de rezolvare, explicând în ce constă eficiența metodei. **(4p.)**
- b. Scrieți programul C/C++ corespunzător metodei descrise la punctul a. **(6p.)**

**Simularea examenului de bacalaureat mai 2013**  
**Proba E. d)**  
**Proba scrisă la INFORMATICĂ**

**BAREM DE EVALUARE ȘI DE NOTARE**  
**(limbajul C/C++)**

Filiera teoretică, profilul real, specializările: matematică – informatică

matematică – informatică intensiv informatică

Filiera vocațională, profilul militar, specializarea matematică – informatică

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct.
- Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- În programele cerute, datele de intrare se consideră corecte, validarea acestora nefiind necesară.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.
- Se vor lua în considerare atât implementările concepute pentru compilatoare pe 16 biți, cât și cele pentru compilatoare pe 32 de biți.

**SUBIECTUL I**

**(30 de puncte)**

1.	b.	4p.	
2.	a.	Răspuns corect: 2	4p.
	b.	Răspuns corect: oricare șir de 10 numere distincte în care ultimul număr trebuie să fie 0 și 5 dintre ele trebuie să aibă cel puțin 3 cifre pare	6p. Se acordă: - numai 1p pentru un răspuns parțial corect care include doar una sau două valori distincte corecte; - numai 2p pentru un răspuns parțial corect care include trei, patru sau cinci valori distincte corecte; - numai 3p pentru un răspuns parțial corect care include șase, șapte sau opt valori distincte corecte.
	c.	Pentru program corect - declarare corectă a tuturor variabilelor - citire corectă - afișare corectă - instrucțiuni de decizie corecte - instrucțiuni repetitive corecte(*) - atribuirii corecte - corectitudinea globală a programului	10p. 1p. 1p. 1p. 2p. 3p. 1p. 1p. (*) Se acordă numai 2p. dacă doar una dintre structuri este corectă.
	d.	Pentru algoritm pseudocod corect - echivalența prelucrării realizate, conform cerinței (*) - corectitudinea globală a algoritmului	6p. 5p. 1p. Se acordă numai 1p. dacă algoritmul are o singură structură repetitivă, principal corectă, dar nu este echivalenta cu cea data. Se va puncta orice formă corectă de structură repetitivă: <b>repetă...până când, repetă...cât timp, execută...cât timp, do...while etc.</b>

**SUBIECTUL al II-lea**

**(30 de puncte)**

<b>1.</b>	<b>a.</b>	<b>4p.</b>	
<b>2.</b>	<b>c.</b>	<b>4p.</b>	
<b>3.</b>	<b>Pentru rezolvare corectă</b> - accesul corect la câmpurile de pe primul nivel al înregistrării - accesul corect la câmpurile de pe al doilea nivel al înregistrării - expresie principal corectă de verificare a proprietății cerute	<b>6p.</b> 2p. 2p. 2p.	
<b>4.</b>	<b>Pentru rezolvare corectă</b> - parcurgerea corectă a celor 2 șiruri căutarea unui caracter într-un șir - identificarea succesiunii corecte de caractere comune - afișarea corectă a literelor comune în ordine crescătoare	<b>6p.</b> 1p. 1p. 2p. 2p.	* dacă literele comune sunt afișate, dar nu în ordine crescătoare * <b>1p</b>
<b>5.</b>	<b>Pentru subprogram corect</b> - declararea corectă a datelor (de tip simplu și tablou) - algoritm principal de determinare a numărului de elemente de pe un contur - completarea corectă a elementelor din matrice - afișarea matricei - corectitudinea globală a subprogramului1)	<b>10p.</b> 1p. 2p. 4p. 2p. 1p.	

**SUBIECTUL al III-lea**

**(30 de puncte)**

1.	d.	4p.	
2.	<b>Răspuns corect</b> 4817 4	6p.	Se acordă numai 3p. dacă o parte din răspuns lipsește sau este incorectă.
3.	<b>Pentru program corect</b> <ul style="list-style-type: none"> <li>- declararea variabilelor</li> <li>- citirea datelor (*)</li> <li>- operații cu fișiere: declarare, pregătire în vederea scrierii, scriere în fișier</li> <li>- scrierea corectă a algoritmului de determinare a cmmdc a două numere</li> <li>- testarea condiției în care cele două numere sunt prime între ele</li> <li>- scrierea tuturor numerelor găsite în fișier</li> <li>- scrierea mesajului corespunzător atunci când nu au fost găsite numerele cerute</li> <li>- corectitudinea globală a programului<sup>1)</sup></li> </ul>	10p. 1p. 1p. 1+1p. 1p. 1p. 2p. 1p. 1p.	(*) Datele de intrare se consideră corecte, validarea acestora nefiind necesară.
4.	<b>a) Pentru program corect</b> <ul style="list-style-type: none"> <li>- operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier</li> <li>- determinarea corectă a unui număr majorant</li> <li>- afișarea mesajului cerut</li> <li>- utilizarea unui algoritm eficient din punct de vedere al timpului de executare (*)</li> <li>- utilizarea unui algoritm eficient din punct de vedere al memoriei utilizate (**)</li> <li>- declarare de variabile, corectitudinea globală a programului<sup>1)</sup></li> </ul> <b>b) Pentru răspuns corect</b> <ul style="list-style-type: none"> <li>- coerența explicării metodei</li> <li>- explicarea unor elemente de eficiență din punct de vedere al timpului de executare</li> <li>- explicarea unor elemente de eficiență din punct de vedere al memoriei utilizate</li> </ul>	6p. 1p. 1p. 1p. 1p. 1p. 1p. 4p. 1p. 2p. 1p.	(*) Punctajul se acordă numai pentru un algoritm liniar. (**) Punctajul se acordă numai pentru un algoritm care utilizează eficient memoria (nu folosește variabile structurate).

<sup>1)</sup> Corectitudinea globală vizează structura, sintaxa și alte greșeli neprecizate în barem.